REMARKS

Amendments have been made to the Title and specification. Claims 1, 6, 13, 18, 25, 30, 39, 44, and 47 have been amended. No new matter has been introduced with these amendments, all of which are supported in the specification as originally filed. Claims 3 - 5, 15 - 17, 27 - 29, and 41 - 43 have been cancelled from the application herein without prejudice. Claims 1 - 2, 6 - 7, 9 - 14, 18 - 19, 21 - 26, 30 - 32, 34 - 37, 39 - 40, 44 - 45, and 47 remain in the application.

I. Objection to the Title

Paragraph 2 of the Office Action dated March 4, 2005 (hereinafter, "the Office Action") states that the Title is objected to as not being descriptive. The Examiner suggests prepending "Method, System, and Computer Program Product" to the current Title. The MPEP states, in §606.11, "Examiner May Require Change in Title", that "This [changing the title] may result in slightly longer titles, but the loss in brevity of title will be more than offset by the gain in its informative value in indexing, classifying, searching, etc.". Applicant respectfully submits that adding the suggested terms to the current Title will in no way provide a "gain in ... informative value [for] indexing, classifying, [or] searching". However, to avoid further delay in passing the application to issuance, Applicant has amended the Title as per the Examiner's suggestion, and the Examiner is therefore respectfully requested to withdraw this objection.

II. Objection to the Claims

Paragraph 9 of the Office Action states that Claims 5 - 6, 17 - 18, and 29 - 30 are objected to because of their numbering. In view of the amendments and cancellations made herein, this objection is rendered moot, and the Examiner is therefore respectfully requested to withdraw the objection.

III. Rejection under 35 U.S.C. §102(b)

Paragraph 12 of the Office Action states that Claims 13 - 19, 21 - 22, 24 - 33, 34 - 35, 37, 39 - 45, and 47 are rejected under 35 U.S.C. §102(b) as being anticipated by Patel et al ("An Efficient Discrete Log Pseudo Random Generator"). Claims 15 - 17, 27 - 29, and 41 - 43 have been cancelled from the application without prejudice, rendering the rejection moot as to those claims. This rejection is respectfully traversed with regard to the remaining claims.

Applicant's independent Claims 13, 25, and 39 (as well as Claim 1, which is rejected under 35 U.S.C. §103 and discussed below) have been amended herein to incorporate limitations from now-cancelled dependent claims, by way of clarification. Applicant's independent claims explicitly specify that the C-bit input value is provided "as an exponent of"(emphasis added) the 1-way function. (See Claim 1, lines 6 - 8, referring to "a length in bits, C, of the input value" and "using the provided input value as an exponent of a 1-way function".) These independent claims further specify that the "base of the modular exponentiation is a fixed generator value" (see Claim 1, line 9).

With reference to Claim 16, which previously contained the limitation pertaining to use of the input value as an exponent, the Office Action cites Patel, page 313, section 5, line 10. However, what is stated therein is use of an exponent x_i and using, as output of an i^{th} step, a subset of the bits of x_i. In particular, Patel states that the lower n - ω(log n) bits of x_i are used as output (while the size of Patel's exponent is n bits). By contrast, Applicant's claimed invention uses a C-bit exponent.

In the Advisory Action dated May 27, 2005 (hereinafter, "the Advisory Action"), the Examiner states that Patel uses a "C"-bit exponent. Applicant respectfully disagrees. When discussing this point, the Advisory Action refers to the above-discussed text on page 313, section 5, line 10 of Patel. This text of Patel will now be discussed in more detail. What is taught therein is that Patel uses an exponent "x_i", and that some "lower" bits are output from the generator function at the ith iteration (when $i > 0$). The number of these lower bits is stated as "the lower n - ω(log n) bits of x_i, except the least significant bit", provided $i > 0$.

However, the number of bits that are output from the generator and the number of bits in the result of the generator are not the same. (If these numbers were the same, then all bits in the result would be output, and the phrase "the *lower* ... bits ..." would have no meaning.) The number of bits in the result of the generator is referred to by Patel as n (Abstract, line 13), and the number of bits output per iteration is referred to as n - c (Abstract, lines 11 - 13). Patel also uses the expression "n - ω(log n)" when describing the number of output bits. See page 313, penultimate line. (That is, "ω(log n)" is also called "c".)

Patel states, by way of example, that n may be 1024 while c may be 128 (Abstract, lines 13 - 14). In other words, the generator result "x_{i-1}" at each iteration may be 1024 bits in length, while "a little less than 900" of those bits are output from the iteration (Abstract, line 14).

Consider the expression of Patel's "new generator" for several example values of *i*, using the algorithm provided on page 313, last paragraph (that is, $x(i+1) = g^{x(i)} \bmod p$), as follows:

$x(0)$ = a seed picked from Z_p^*

$x(1) = g^{x(0)} \bmod p$ -- $i = 0$, therefore no output

$x(2) = g^{x(1)} \bmod p$ -- $i = 1$, output lower (n - c) bits of $x(1)$

$x(3) = g^{x(2)} \bmod p$ -- $i = 2$, output lower (n - c) bits of $x(2)$

$x(4) = g^{x(3)} \bmod p$ -- $i = 3$, output lower (n - c) bits of $x(3)$

$x(5) = g^{x(4)} \bmod p$ -- $i = 4$, output lower (n - c) bits of $x(4)$

$x(6) = g^{x(5)} \bmod p$ -- $i = 5$, output lower (n - c) bits of $x(5)$

It may be that Patel actually meant "... when $(i + 1) > 0$, output the lower ... bits of $x(i + 1)$, ..." -- because otherwise, Patel is outputting bits from the exponent rather than bits of the result of the generator. Accordingly, the example expressions may be rewritten as follows:

$x(0)$ = a seed picked from Z_p^* -- $i + 1 = 0$, therefore no output

$x(1) = g^{x(0)} \bmod p$ -- $i = 0$, output lower (n - c) bits of $x(1)$

$x(2) = g^{x(1)} \bmod p$ -- $i = 1$, output lower (n - c) bits of $x(2)$

$x(3) = g^{x(2)} \bmod p$ -- $i = 2$, output lower (n - c) bits of $x(3)$

$x(4) = g^{x(3)} \bmod p$ -- $i = 3$, output lower (n - c) bits of $x(4)$

$x(5) = g^{x(4)} \bmod p$ -- $i = 4$, output lower (n - c) bits of $x(5)$

$x(6) = g^{x(5)} \bmod p$ -- $i = 5$, output lower (n - c) bits of $x(6)$

At any rate, what can be seen by these example expressions for several iterations of Patel's generator is that the exponent being used in each iteration is specified as the *entire* output of the prior iteration. For example, when $i = 2$ in the second set of expressions provided above, Patel teaches that "a little less than 900 bits" of $x(3)$, or "the lower n - ω(log n) bits" of $x(3)$, are output from the generator. However, the algorithm does not state that only "ω(log n) bits" (i.e., "c" bits) of $x(3)$ are used as the exponent when computing $x(4)$. Instead, the algorithm specifies that the exponent used when computing $x(4)$ is $x(3)$ -- and as noted above, the length of $x(3)$ is n bits. This is distinct from Applicant's claimed invention.

The Advisory Action refers to page 314, section 5.1 of Patel as teaching that the "input exponent when using short exponents was ω(log n) bits, or "C" bits". Applicant respectfully disagrees: section 5.1 does not specify Patel's algorithm. The algorithm was specified in section 5 on page 313, which has been discussed above. Section 5.1, on the other hand, is a "proof of security" of Patel's algorithm. In this proof of security, Patel uses a different value "g' mod p" (that is, g ** s), where the exponent s has ω(log n) bits. In the second paragraph, last sentence, of section 5.1, Patel again states that it is s -- that is, the exponent used in the algorithm for proving the security of the generator -- that has length ω(log n) bits, referring to this length as a "short exponent". However, the length of the exponent used in the proof of security is *not* the length of the exponent used in Patel's generator.

It should also be noted that Patel explicitly refers to the size of his generator's exponents as "large". See section 7.1, "Improving Efficiency of Computations", in Patel's Appendix. In line 3 of the first paragraph, Patel again presents his generator algorithm using the *entire* output of a prior iteration as the exponent in each next iteration, and in lines 3 - 4, refers to the output bits of the generator's iterations. In the next paragraph, lines 1- 2, Patel states "Although the number of bits generated per iteration [of the generator] is large, each iteration involves a large exponent and this could impact on the speed of the generator" (emphasis added). This use of large exponents is distinct from Applicant's claimed invention.

Patel continues, in this second paragraph of section 7.1, by noting a possible alternative approach where the length of the exponent for the generator could be shortened to *s(i)*, where *s(i)* is the "leading ω(log *n*) bits of [the generator's result] *x(i)*", and then stating that this approach "will ensure that at each stage we are using short exponents ...". However, Patel continues by stating that this alternative approach "raises some interesting questions", and -- as the answer to Question 10, which asks whether using short exponents will impact the security of the generator -- states that "when we restrict our exponents [to the "short" exponents comprising ω(log *n*) bits of the generator's output] we no longer have a permutation ... [and therefore] the simple construction used here [i.e., the simple algorithm that can output nearly 900 bits per iteration] is inapplicable" (emphasis added). See page 316, section 7.1, paragraph 2, last sentence and paragraph 3, lines 1 - 2.

Accordingly, this text on page 316 teaches away from using only C bits of the generator's output as the exponent of the next iteration.

Because Patel's generator function uses n-bit exponents, stating that these are "large" exponents, and in his alternative approach, Patel teaches away from using the C-bit ("substantially shorter") exponents of Applicant's claimed invention, Applicant respectfully submits that his independent Claims 1, 13, 25, and 39 are patentable over the teachings of Patel. Dependent Claims 14, 18 - 19, 21 - 22, 24, 26, 30 - 33, 34 - 35, 37, 40, 44 - 45, and 47 are therefore deemed patentable over the reference as well. The Examiner is therefore respectfully requested to withdraw the §102 rejection.

IV. Rejection Under 35 U.S.C. §103(a)

Paragraph 30 of the Office Action states that Claims 23 and 36 are rejected under 35 U.S.C. §103(a) as being unpatentable over Patel in view of Schneier ("Applied Cryptography"). Paragraph 31 - 32 of the Office Action state that Claims 1 - 7 and 9 - 12, respectively, are also rejected using these references. Claims 3 - 5 have been cancelled from the application without prejudice, rendering the rejection moot as to those claims. These rejections are respectfully traversed with respect to the remaining claims.

Applicant's independent Claims 1, 13, 25, and 39 have been discussed above, and as has been demonstrated, Patel does not anticipate these independent claims. Accordingly, Patel cannot be combined with Schneier (assuming, *arguendo,* that such combination could be made, and that one of skill in the art would be motivated to attempt it) to render dependent Claims 2, 6 - 7, 9 - 12, 23, and 36 unpatentable. The Examiner is therefore respectfully requested to withdraw the §103 rejection.

V. Conclusion

Applicant respectfully requests reconsideration of the pending rejected claims, withdrawal of all presently outstanding objections and rejections, and allowance of all remaining claims at an early date.

Respectfully submitted,

Marcia L. Doubet

Marcia L. Doubet
Attorney for Applicant
Reg. No. 40,999

Customer Number for Correspondence: 43168
Phone: 407-343-7586
Fax: 407-343-7587